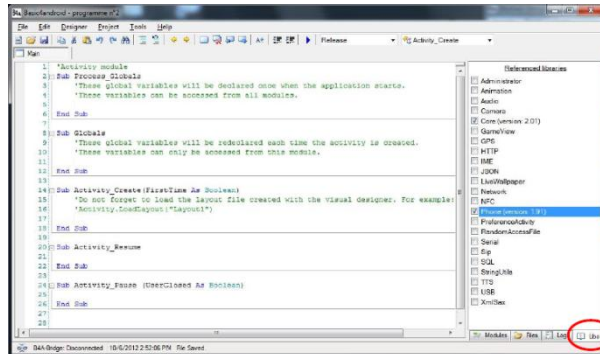


Programme

Ce programme va vous permettre de mettre en oeuvre un des capteurs de la tablette : l'accéléromètre (ou l'inclinomètre).

- 1- Créez un nouveau projet B4A que vous sauvegarderez dans un dossier **programme2** et que vous nommerez « **Programme n2** ».
- 2- Pour pouvoir mettre en oeuvre l'accéléromètre il va falloir utiliser une librairie particulière. B4A propose par défaut un corpus de librairies, que l'on peut facilement étendre en les téléchargeant sur Internet. Pour les atteindre il faut utiliser l'onglet **Libs**. Validez la librairie **Phone**



La description complète de la librairie est ici :

<http://www.basic4ppc.com/android/help/phone.html>

Vous allez utiliser l'objet **PhoneAccelerometer** de cette librairie et une propriété de l'objet : **StartListening** ainsi que son évènement associé **AccelerometerChanged (X As Float, Y As Float, Z As Float)**.

- 3- Complétez les lignes suivantes :

```
1 'Activity module
2 Sub Process_Globals
3     'These global variables will be declared once when the application starts.
4     'These variables can be accessed from all modules.
5     Dim capteurACC As PhoneAccelerometer
6 End Sub
7
8 Sub Globals
9     'These global variables will be redeclared each time the activity is or
10    'These variables can only be accessed from this module.
11
12 End Sub
13
14 Sub Activity_Create(FirstTime As Boolean)
15     'Do not forget to load the layout file created with the visual designer
16     capteurACC.StartListening("inclinaison")
17 End Sub
18
19 Sub Activity_Resume
20
21 End Sub
22
23 Sub Activity_Pause (UserClosed As Boolean)
24
25 End Sub
26 Sub inclinaison_accelerometerchanged(X As Float,Y As Float,Z As Float)
27
28 end sub
```

- 4- Dans le Designer placez trois **Label** que vous nommerez **X_Label**, **Y_Label** et **Z_Label**. Fermez le Designer en ayant au préalable utilisé la commande **Tools-Generate Members** et sauvegardé votre feuille sous le nom **testACC**.
- 5- Constatez les changements dans votre programme et rajoutez la commande permettant de charger au démarrage votre feuille :

```
7
8 Sub Globals
9     'These global variables will be redeclared each time the activity is created.
10    'These variables can only be accessed from this module.
11    Dim X_Label As Label
12    Dim Y_Label As Label
13    Dim Z_Label As Label
14 End Sub
15
16 Sub Activity_Create(FirstTime As Boolean)
17     'Do not forget to load the layout file created with the visual designer. For e
18     Activity.LoadLayout("testACC")
19     CapteurACC.startListening("Inclinaison")
20 End Sub
21
22 Sub Activity_Resume
23
24 End Sub
25
```

On veut renvoyer sur la propriété **X_Label.Text** la valeur **X** retournée par l'accéléromètre. Le problème est que cette propriété attend une chaîne de caractères (**String**) alors que **X** est un nombre en virgule flottante (**float**). Il va donc falloir réaliser une modification de format de variable.

Pour cela B4A propose plusieurs fonctions. Celle nous permettant de résoudre notre problème est : **NumberFormat (Number As Double, MinimumIntegers As Int, MaximumFractions As Int) As String**.

Cette fonction renvoie une chaîne de caractères à partir du nombre **Number** (qui peut aussi être un **float**), composée d'au moins **MinimumIntegers** chiffres pour la partie entière de **Number**, et d'un maximum **MaximumFractions** chiffres pour la partie décimale.

- 6- Aussi modifiez votre programme pour que la chaîne de caractère **X_Label.Text** soit réalisée à partir de la variable **X** en ayant une partie entière au minimum d'un chiffre, et une partie décimale au maximum de 2 chiffres.
- 7- Histoire de bien se familiariser avec cette fonction, faites en sorte que **Y_Label.Text** s'écrive 00.000 et **Z_Label.Text** uniquement en entier 00.

```
29 Sub inclinaison_accelerometerchanged(X As Float,Y As Float,Z As Float)
30     X_Label.Text=NumberFormat(X,1,2)
31     Y_Label.Text=NumberFormat(Y,2,3)
32     Z_Label.Text=NumberFormat(Z,2,0)
33 End Sub
34
```

Remarque : B4A est capable de réaliser seul des conversions de format. Ainsi on peut remplacer la ligne **X_Label.Text=NumberFormat(X,1,2)** par **X_Label.Text=X**. Cela fonctionne parfaitement, par contre il n'y a plus de formatage de l'écriture entière/décimale

Le programme est maintenant terminé, on va pouvoir le tester. Hélas, pas de test possible sur le simulateur, il faut transférer le programme dans la tablette.

Pour cela on va utiliser le Bridge B4A.

- 8- Lancez l'application **B4A-Bridge** sur la tablette (elle normalement être installée).
- 9- Repérez l'adresse IP de la tablette (192.168.....).
- 10- Sur B4A, lancez la commande **Tools – B4A Bridge – Connect Wireless** dans B4A.
- 11- Saisissez l'adresse IP relevée précédemment.
- 12- Testez votre programme en cliquant sur la flèche bleue (comme avec le simulateur).
- 13- Sur la tablette acceptez l'installation et ouvrez votre programme.
- 14- Vous devez constater que les inclinaisons varient de -10 à +10 (cela peut varier d'une tablette à l'autre) , mais que la lecture est difficile, puisque le basculement de la tablette entraine un basculement du sens de lecture sur l'écran. Pour cela B4A vous permet de verrouiller votre application pour qu'elle ne fonctionne que dans un sens en utilisant le menu **Project-Orientations supported**. Dans le même menu on trouve **Chose Icon** permettant de sélectionner un fichier image utilisé comme icône sur votre tablette.

Projet

Vous allez créer un projet permettant de déplacer une image sur la tablette en X et Y et ce en fonction de l'inclinaison de celle-ci.

- 1- Créez un nouveau projet B4A que vous sauvegarderez dans un dossier **projetAcc** et que vous nommerez « **projetAcc** ».
- 2- Allez sur Internet pour récupérer une image, celle que vous voulez. Ce sera l'objet que vous déplacerez sur la tablette.
- 3- Ouvrez le Designer et placez-y votre image.
- 4- Réduisez sa taille.
- 5- Déplacez la sur l'écran et notez les valeurs X_{max} , X_{min} , Y_{max} et Y_{min} avant qu'elle ne sorte de l'écran.
- 6- Fermez le Designer en ayant au préalable généré les membres.

On va pour le moment gérer uniquement le déplacement en X.

La première idée consiste à utiliser l'inclinaison en X et si elle dépasse deux limites à modifier la position en X de l'image. On a alors ce programme :

```
1 'Activity module
2 Sub Process_Globals
3     'These global variables will be declared once when the application starts.
4     'These variables can be accessed from all modules.
5     Dim capteurACC As PhoneAccelerometer
6 End Sub
7
8 Sub Globals
9     'These global variables will be redeclared each time the activity is created.
10    'These variables can only be accessed from this module.
11    Dim ImageView1 As ImageView
12 End Sub
13
14 Sub Activity_Create(FirstTime As Boolean)
15     'Do not forget to load the layout file created with the visual designer. For example:
16     Activity.LoadLayout("testACC")
17     capteurACC.StartListening("inclinaison")
18 End Sub
19
20 Sub Activity_Resume
21
22 End Sub
23
24 Sub Activity_Pause (UserClosed As Boolean)
25
26 End Sub
27 Sub inclinaison_accelerometerchanged(X As Float,Y As Float,Z As Float)
28     If X>2 Then ImageView1.Left=ImageView1.Left-1
29     If X<-2 Then ImageView1.Left=ImageView1.Left+1
30 End Sub
31
```

7- Saisissez-le

8- Cherchez sur le site Internet de B4A la syntaxe et le fonctionnement des structures en **If Then**.

9- Testez votre programme sur la tablette en utilisant le Bridge B4A

Vous devez constater que cela ne fonctionne pas parfaitement : le déplacement est saccadé, et l'image peut même se bloquer alors que la tablette est encore inclinée. Cela est parfaitement normal puisque la fonction **inclinaison_accelerometerchanged** n'est appelée que si l'utilisateur change l'inclinaison. Donc s'il reste dans une position, même inclinée, la fonction n'est pas appelée et l'image n'est donc pas déplacée.

La solution consiste à sauvegarder l'inclinaison X et à déplacer la tablette tant que cette inclinaison est au-delà des limites (ici -2 et 2).

Pour cela on va commencer par faire deux modifications.

10- La première consiste à créer une variable **xsave** de type **float** :

```
1 'Activity module
2 Sub Process_Globals
3     'These global variables will be declared once when the application starts.
4     'These variables can be accessed from all modules.
5     Dim capteurACC As PhoneAccelerometer
6     Dim xsave As Float
7 End Sub
```

11- Puis à lui donner la valeur de X lorsque la position de la tablette change :

```
28 Sub inclinaison_accelerometerchanged(X As Float,Y As Float,Z As Float)
29     xsave=X
30 End Sub
```

Maintenant il faut que notre programme « interroge » régulièrement la valeur de **xsave** pour que si elle est au-delà des limites, on déplace l'image.

IHM : Mise en œuvre d'un accéléromètre sous B4A

Le problème est que B4A propose des fonctions qui sont exécutées suite à un évènement : un appui sur un bouton, une inclinaison.... Il faut donc que l'on trouve un objet qui déclenche régulièrement et seul, des évènements. C'est le cas de l'objet **Timer**.

Celui-ci n'est pas disponible dans le Designer puisqu'il n'a pas d'interaction possible avec l'utilisateur. Il va donc falloir travailler directement avec le programme en VB.

Cet objet possède trois paramètres :

Initialize : initialisation du timer en définissant le nom de l'évènement associé à l'objet et l'intervalle en ms où le timer déclenche un évènement.

Syntaxe : nom_du_timer.Initialize(« nom_de_l'évènement » as String, Intervalle as Long)

Exemple : Timer1.initialize(« Timer1 »,1000)

Interval : pour redéfinir l'intervalle en ms où le timer déclenche un évènement.

Syntaxe : nom_du_timer.Interval=Intervalle as Long

Exemple : Timer1.Interval=1000

Enabled : valide ou non les fonctionnalités du timer

Syntaxe : Timer1.Enabled=True ou False

Un seul évènement est associé : **Tick** lorsque le timer atteint son intervalle de temps

Sub Timer1_Tick

12- La première étape consiste donc à déclarer l'objet **Timer**. Modifiez le programme

```
9 Sub Globals
10     'These global variables will be redeclared each time the activity is created.
11     'These variables can only be accessed from this module.
12     Dim ImageView1 As ImageView
13     Dim Montimer As Timer
14 End Sub
```

13- Au lancement de votre programme il faut initialiser le **Timer** :

```
15
16 Sub Activity_Create(FirstTime As Boolean)
17     'Do not forget to load the layout file created with the visual designer. For example:
18     Activity.LoadLayout("testACC")
19     capteurACC.StartListening("inclinaison")
20     montimer.Initialize("top",10)
21     montimer.Enabled=True
```

14- Ensuite, il faut créer la fonction associée à l'objet :

```
34
35 Sub top_tick
36     If xsave>2 Then ImageView1.Left=ImageView1.Left+1
37     If xsave<-2 Then ImageView1.Left=ImageView1.Left-1
38 End Sub
```

15- Testez votre programme.

16- Rajoutez maintenant un test interdisant le déplacement au-delà des limites X de l'écran (indice : le test en **If** peut prendre deux conditions...)

17- Le déplacement est fluide, mais pas très rapide. Trouvez une solution pour l'accélérer (au moins deux solutions possibles, une marchant mieux que l'autre)

18- Sur le même principe gérez maintenant l'inclinaison en Y.